基于中心化相似度矩阵的词向量方法 *

徐　帆，王裴岩，蔡东风

(沈阳航空航天大学 人机智能研究中心，沈阳 110136)

摘　要：词向量使用低维稠密向量表示词，通过向量运算能够反映词间关系，被广泛应用于自然语言处理任务。对基于矩阵分解的词向量方法进行了研究，发现降维前相似度矩阵质量与词向量质量存在线性相关性，提出了一种基于中心化相似度矩阵的方法。该方法使得相似（不相似或弱相似）词间的相似程度相对增强（减弱）。在 WS-353 和 RW 数据集的词语相似性实验中验证了所提出方法的有效性，两个数据集下词向量质量最高提升 0.2896 和 0.1801。中心化能够提升降维前相似度矩阵质量，进而提升词向量质量。

关键词：词向量；中心化；相似度矩阵

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2017.08.0721

Method of word vector based on centring similarity matrix

Xu Fan, Wang Peiyan, Cai Dongfeng

(*Human-computer Intelligence Research Center*, *Shenyang Aerospace University*, *Shenyang* 110136, *China*)

Abstract: The word vector, which represents word by a low dimensional dense vector. The relationships between words are denoted by vector operations. Hence it is broadly applied in tasks of natural language processing. The method of word vector based on matrix factorization is studied. It found that there is a linear correlation between the quality of no dimension reduction matrix and the quality of word vector. Furthermore, it derived a method of the word vector, which based on a kind of centring similarity matrix. This method makes the similarity between similar (dissimilar or weakly similar) words relatively enhanced (weakened) . In the word similarity experiments of WS-353 and RW datasets, the effectiveness of the proposed method is verified. The highest quality of the word vectors among the two datasets is 0.2896 and 0.1801. Centralization can improve the quality of similarity matrix, moreover it can improve the quality of word vector.

Key Words: the word vector; centralization; similarity matrix

0　引言

词向量[1,2]可以从大量的未标注语料中提取词的语义和句法信息，因此引起了广泛的关注[3]。词向量将语料库中的每个词表示为一个低维实数向量，建立离散词与实数域特征向量之间的映射，词间语义越相似，其向量表示越接近[4]。词向量可以用于计算词间相似度，也可以作为特征直接应用于词义消歧[4]、文本分类[5,6]、词性标注[1,7]、情感分析[6,8]等自然语言处理任务。

词向量的方法可分为基于矩阵分解的方法和基于预测的方法[9,10]。基于矩阵分解的方法源自词的分布假设，即词的上下文内容相似，则词本身的含义也相似[9]。该方法可以追溯到 LSA（latent semantic analysis）[11,12]，通过分解词-文档矩阵[10]而获得词向量。而目前普遍使用的是词-上下文共现[9]权重为词的语义空间分布矩阵赋值，从而获得词向量。这种表示方法最早源自于 HAL（hyperspace analogue to language）[13]，利用滑动窗口构造词—上下文共现矩阵。此后，基于矩阵分解的词向量方法都是使用词—上下文共现矩阵进行构造的。基于预测的词向量方法源自神经网络模型[9]。Mikolov 等人[14,15]提出 CBOW（continue bag of words）和 Skip-gram 两种基于预测的词向量方法，因为其训练得到的词向量有很好的语义特性从而得到了广泛的关注[9]。Levy 等人[16]对基于矩阵分解的词向量方法与 skip-gram 在词语相似性任务上进行了细致的分析，尝试多种不同的参数，发现两种方法在大多数参数设置的方法中能够达到相近效果。在 Skip-gram 被广泛使用后，研究者们开始关注其与基于矩阵分解的词向量方法之间的关系，并重点研究其理论的可解释性。Levy 等人[17]表明 SGNS（skip-gram negative sampling）的训练方法可看做加权矩阵分解，这种加

基金项目：辽宁省自然科学基金计划重点项目（20170540705）；国家自然科学基金资助项目（61403262）

作者简介：徐帆（1993-），女，辽宁沈阳人，硕士研究生，主要研究方向为人工智能与自然语言处理；王裴岩（1983-），男（通信作者），讲师，博士，主要研究方向为机器学习、信息抽取（wangpy@sau.edu.cn）；蔡东风（1958-），男，教授，博士，主要研究方向为机器学习、人工智能、自然语言处理．

权矩阵分解等价于隐式分解 SPPMI（shift positive PMI）矩阵。Li 等人[18]也发现 SGNS 等价于一种词-上下文共现矩阵分解，并根据这种等价性引入监督信息，在词类比任务中给定 10%的训练数据就能取得 9%的性能提升。Pennington 等人[10]利用 Skip-gram 能够挖掘出词间线性关系的特性以及 SGNS 与矩阵分解的等价关系提出了 Glove 模型，发现该模型在词类比任务中表现较好。由于 skip-gram 与基于矩阵分解的词向量方法都能够看成是针对词—上下文共现矩阵进行的研究，并且两者得到的词向量的质量具有可比性，因此许多研究者再次开始了基于矩阵分解的词向量方法的研究。其中，最具有代表性的是 Hellinger PCA（hellinger principal component analysis，HPCA）[19]方法。文献[19,20]利用 HPCA 的方法获得词向量，首先使用条件概率为词—上下文共现矩阵进行赋值，然后使用 Hellinger 距离对共现矩阵的每两行计算相似度，得到相似度矩阵，最后对相似度矩阵进行降维得到词向量，发现在词语相似性任务和词类比任务中效果较好。

本文基于文献[19,20]中的算法过程对基于矩阵分解的词向量方法进行了研究，发现降维前的相似度矩阵直接影响词向量的质量，通过 Pearson 相关系数验证了两者之间具有较强的线性相关性。并且，提出了一种基于中心化相似度矩阵的词向量方法。该方法通过对相似度矩阵中心化，使得相似词间的相似程度相对增加，不相似或弱相似的词间的相似程度相对减弱。在词语相似性任务上，验证了该方法的有效性，中心化相似度矩阵获得的词向量的质量明显好于非中心化相似度矩阵获得的词向量的质量。

1　基于矩阵分解的词向量方法

基于矩阵分解的词向量方法通过上下文分布的共现情况描述词的语义，具体步骤为[9,19,20]：首先构建词—上下文共现矩阵 C；然后对 C 的每两行进行相似度计算得到相似度矩阵 A；最后对 A 进行降维得到词向量矩阵 E。

1.1　词-上下文共现矩阵的构建

词—上下文共现矩阵 C 的每个元素表示词 w_i 与上下文词 c_j 的共现权重 $t(w_i,c_j)$。V 为待表示词的数量，D 为上下文词的数量。因此，C 为 $V\times D$ 的矩阵，每一行为词 w_i 基于上下文词 c_j 的向量表示，即

$$C_{i,\cdot}=[t(w_i,c_1),t(w_i,c_2),\ldots,t(w_i,c_D)]$$

$t(w_i,c_j)$ 的计算方法有词频（TF）、点互信息（PMI）[17,21]和条件概率（CP）[20]。文献[17]中提出将 PMI 的方法改为 PPMI，并使用 SPPMI 方法得到与 Skip-gram 模型等价的结论。文献[20]与 Glove 模型则使用条件概率计算 $t(w_i,c_j)$。权重 $t(w_i,c_j)$ 具体计算方法如表 1 所示，其中 $\#(w_i,c_j)$ 表示 w_i 与 c_j 共现的次数，$\#(w_i)$ 和 $\#(c_j)$ 分别表示 w_i 和 c_j 在语料库中出现的次数，N 为语料库中词的总数。当 w_i 和 c_j 未共现时，$t^{PMI}(w_i,c_j)=\log 0=-\infty$。因此本文规定，当 $\#(w_i,c_j)=0$ 时，$t^{PMI}(w_i,c_j)=0$。

表 1　权重 $t(w_i,c_j)$ 的计算方法

方法名称	计算方法
词频	$t^{TF}(w_i,c_j)=\#(w_i,c_j)$
点互信息	$t^{PMI}(w_i,c_j)=\log(\frac{\#(w_i,c_j)\times N}{\#(w_i)\times\#(c_j)})$
条件概率	$t^{CP}(w_i,c_j)=\frac{\#(w_i,c_j)}{\#(c_j)}$

1.2　相似度矩阵的构建

对词-上下文共现矩阵 C 中的每两行向量 $C_{i,\cdot}$ 和 $C_{j,\cdot}$ 做相似度计算得到对称的相似度矩阵 A，$A_{i,j}$ 表示 w_i 和 w_j 的相似度 $sim(w_i,w_j)$。$sim(w_i,w_j)$ 的计算方法有余弦相似度[22]、欧氏距离[13]、Hellinger 距离[19,20]等。本文使用余弦相似度和 Hellinger 距离两种方法对 C 进行相似度计算。

余弦相似度是常见的相似度计算方法，如式（1）所示。

$$sim^{\cos}(w_i,w_j)=\frac{C_{i,\cdot}\cdot C_{j,\cdot}}{|C_{i,\cdot}|\times|C_{j,\cdot}|} \tag{1}$$

Hellinger 是用来度量两个离散概率分布的相似度。因此，需要对待表示词的向量做归一化处理，处理后的向量为 $P_{i,\cdot}$：

$$P_{i,\cdot}=[\frac{t(w_i,c_1)}{\sum_{k=1}^{D}t(w_i,c_k)},\frac{t(w_i,c_2)}{\sum_{k=1}^{D}t(w_i,c_k)},\ldots,\frac{t(w_i,c_D)}{\sum_{k=1}^{D}t(w_i,c_k)}]$$

则 Hellinger 距离的计算公式如式（2）所示。

$$sim^{H}(w_i,w_j)=\frac{1}{\sqrt{2}}\sqrt{\sum_{k=1}^{D}(\sqrt{p_{i,k}}-\sqrt{p_{j,k}})^2} \tag{2}$$

1.3　矩阵分解

由于相似度矩阵 A 是对称方阵，因此本文使用特征值分解对 A 进行降维，得到词向量矩阵 E。特征值分解是只保留前 d 个特征值对应的特征向量达到降维的目的，特征值越大，对应的特征向量方向上包含的信息量越多。通过特征值分解得到前 d 个特征值对应的特征向量，即对应了该矩阵最主要的 d 个变化方向，利用这 d 个变化方向就可以近似这个矩阵，实现降维。

降维可以看成是将 A 映射至一个低维空间，得到 A 的映射矩阵 $\hat{A}$，使得 $\hat{A}$ 与 A 的差值尽量小。对 A 进行特征值分解，如式（3）所示，其中 Q 为 A 的特征向量矩阵，Σ 为特征值矩阵。

$$A=Q\Sigma Q^{T} \tag{3}$$

因此，可以通过对 Σ 进行排序，选择前 d 个特征值所对应的特征向量对 A 进行表示。

本文利用文献[12]中 Caron 等人提出的 $Q\Sigma\rightarrow Q\Sigma^{p}$ 形式进行降维。Levy 等人提出最佳的 p 值应为 $p<1$[23]。文献[24]中更提出 $p=0.5$ 时结果最好。因此，我们对 A 分解出的特征值进行排序，使用前 d 个特征值对应的特征向量与特征值的 p 次幂的乘积进行降维，p 值为 0.5，如式（4）所示。

$$E=Q_d\Sigma_d^{0.5} \tag{4}$$

1.4　基于矩阵分解的词向量方法构造

通过组合上述 3 种权重计算方法和 2 种相似度计算方法，构造出 5 种基于矩阵分解的词向量方法，具体如表 2 所示，降维都采用上述特征值分解方法。其中 PMI 计算结果有负值，因此不将 PMI 与 Hellinger 距离进行搭配。

表 2　基于矩阵分解的词向量方法

方法名称	权重计算方法	相似度计算方法
TCE	词频	余弦相似度
PCE	点互信息	余弦相似度
GCE	条件概率	余弦相似度
THE	词频	Hellinger 距离
GHE	条件概率	Hellinger 距离

2　基于中心化的相似度矩阵优化

本节将介绍基于中心化的相似度矩阵优化方法。本文中使用的 Hellinger 距离与余弦相似度两种相似度计算方法均可看作一种内积运算。根据核函数的定义，两种相似度计算方法均可视为一种核函数。核函数[25,26]是两样本点在特征空间中的内积，决定数据在特征空间中的分布。其形式描述如式（5）所示。其中 X 为输入空间，$X \subset R^n$，H 为特征空间。

$$k(x,z) = <\varphi(x),\varphi(z)>, \varphi: X \to H \tag{5}$$

文献[26]提出中心化核函数的方法，使用 K_α 代表中心化核函数，公式如式（6）所示。

$$\begin{aligned} k_\alpha(x_i,x_j) &= <\varphi(x_i) - \frac{1}{n}\sum_i^n \varphi(x_i), \varphi(x_j) - \frac{1}{n}\sum_j^n \varphi(x_j)> \\ &= k(x_i,x_j) - \frac{1}{n}\sum_{i=1}^n k(x_i,x_j) - \frac{1}{n}\sum_{j=1}^n k(x_i,x_j) + \frac{1}{n^2}\sum_{i,j=1}^n k(x_i,x_j) \end{aligned} \tag{6}$$

如果在特征空间中数据远离原点，那么核矩阵中的元素将几乎相等，该核矩阵是病态核矩阵[25]。中心化可以消除由于样本远离原点而产生的病态核矩阵的问题[25]。依据该视角，基于矩阵分解的词向量方法通过相似度函数构造了特征空间。中心化使得词在特征空间中围绕原点，可使得相似度矩阵中的元素间差别较大，有效区分词间相似程度。

对相似度矩阵 A 进行中心化的具体方法为：将 A 中每个元素减去其所在行、列的平均值并加上矩阵所有元素的平均值得到中心化后的相似度矩阵 $\tilde{A}$，计算公式如（7）和（8）所示。I 为单位矩阵，M 为 $V \times V$ 的全 1 矩阵，即 $[M]_{i,j} = 1$。

$$\tilde{A}_{i,j} = A_{i,j} - \frac{1}{V}\sum_{j=1}^V A_{i,j} - \frac{1}{V}\sum_{i=1}^V A_{i,j} + \frac{1}{V^2}\sum_{i=1}^V\sum_{j=1}^V A_{i,j} \tag{7}$$

$$\tilde{A} = \left(I - \frac{1}{V}M\right) A \left(I - \frac{1}{V}M\right)^T \tag{8}$$

为进一步说明中心化的作用，以“lobster”（龙虾）为例，选择了“seafood”（海鲜），“eye”（眼睛），“glass”（玻璃），“boy”（男孩）与“shore”（海岸）这 5 个词，并观察在中心化前后上述词的相似度变化情况，中心化前后相似度的值如表 3 所示。观察中心化前后的相似度值不难发现，中心化前每组词的相似度较为接近，分布在 0.27 与 0.35 之间。而在中心化之后则体现出了显著的差异。例如“lobster”与“eye”，“lobster”与“seafood”两组词，在中心化之前，相似度分别为 0.3238 与 0.3272。两组词具有较为相近的相似度，即“eye”与“seafood”同“lobster”的语义关系相同，这显然不合理。在中心化之后，“lobster”与“eye”的相似度为-0.0017，“lobster”与“seafood”的相似度为 0.2868。“eye”与“lobster”从相似变为不相似，两组词的相似度差异明显。由此可见，中心化后能使得相似词的相似程度相对增强，不相似或弱相似词的相似程度相对减弱，相似度矩阵更加合理。

表 3　中心化前后 5 组词相似度值对比结果

词对	中心化前	中心化后
（lobster，eye）	0.3238	-0.0017
（lobster，seafood）	0.3272	0.2868
（lobster，glass）	0.3079	0.0128
（lobster，boy）	0.3191	-0.0165
（lobster，shore）	0.2710	0.0340

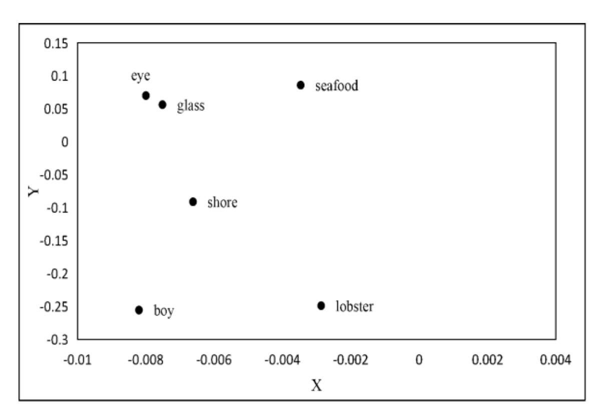


图 1　中心化前“lobster”等词的二维词向量的分布情况

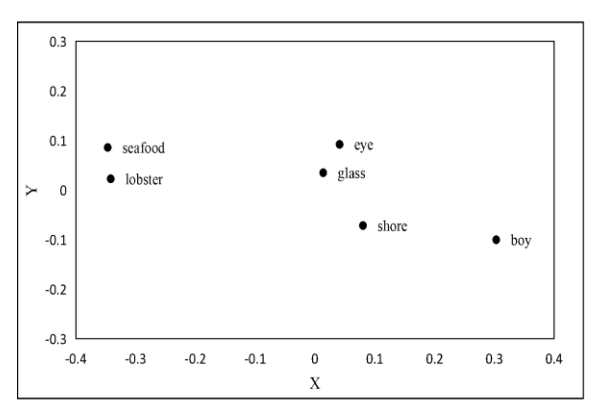


图 2　中心化后“lobster”等词的二维词向量的分布情况

图 1 与 2 分别展示了上述 6 个词中心化前后 2 维词向量的空间分布。从图 1 中能够看出在中心化前，所有的词聚集在原点左侧区域。大多数词与“lobster”的距离几乎相同，看不出哪些词与其更相似或更不相似。在中心化后，所有的词均围绕

原点，并且可明显看出“lobster”与“eye”的距离大于“lobster”与“seafood”在距离。由此可见，中心化能够使相似的词在空间中的分布相对较近，不相似或弱相似的词在空间中的分布相对较远，词向量更加合理，词能够得到更好的词向量表示。

3 实验

3.1 实验设置

本文使用 2015 年维基百科英文语料库[1]作为训练集，共 3,991,454 篇。在预处理过程中将训练集中的词处理成小写并去除标点符号。由于训练集的词汇量较大，为方便计算，本文选择去除部分高频词与低频词，使用训练集中频率居中（10^{-6}~10^{-5}）的词构建词表，共计有 30,946 个词。词表的选择可参考文献[20]。

目前，评价词向量的质量是通过计算两个词的相似度来进行的，因此在词语相似性任务中对本文训练出的词向量进行评价。使用余弦相似度及内积相似度两种方法判断两个词向量间的相似程度。余弦相似度是常用的计算两个词向量相似度的计算方法。由式（4）对词向量做内积相似度相当于对相似度矩阵的近似矩阵进行还原，因此可便于揭示相似度矩阵与词向量质量的相关性。使用 Spearman 相关系数[27]对余弦相似度和内积相似度两种方法计算出的相似度与人工标注的词相似度数据集进行评价。公开的人工标注词相似度数据集选择 WS-353[28]和 RW[29]，WS-353 是由常见的 353 对词组成，主要标注了名词、动词及形容词间的相似度[9]，RW 是由斯坦福稀有词汇或词法复杂的词对构成。

对比方法选择所构建的 5 种基于矩阵分解的词向量方法（表 2），以及 Skip-gram 和 Glove。上述所有模型均在 5 窗口及 100 维的条件下训练词向量[10,16,30]。

3.2 实验结果与分析

3.2.1 词语相似性实验

表 4 至表 7 分别展示了在不同数据集与相似度计算方法下的实验结果。

表 4 WS-353 数据集点积相似度的词向量质量

中心化情况	TCE	PCE	GCE	THE	GHE
中心化后	0.6325	0.5305	**0.6401**	0.6111	0.6136
中心化前	0.5397	0.2409	0.5614	0.3955	0.4057
性能提升	0.0928	**0.2896**	0.0787	0.2156	0.2079

表 5 RW 数据集点积相似度的词向量质量

中心化情况	TCE	PCE	GCE	THE	GHE
中心化后	**0.3098**	0.2117	0.2989	0.2431	0.2521
中心化前	0.1731	0.0316	0.1731	0.0648	0.0778
性能提升	0.1367	**0.1801**	0.1258	0.1783	0.1743

表 6 WS-353 数据集余弦相似度的词向量质量

中心化情况	TCE	PCE	GCE	THE	GHE
中心化后	0.6253	0.4865	**0.6257**	0.5787	0.5787
中心化前	0.5556	0.2691	0.5544	0.3670	0.3828
性能提升	0.0697	**0.2174**	0.0713	0.2117	0.1959

表 7 RW 数据集余弦相似度的词向量质量

中心化情况	TCE	PCE	GCE	THE	GHE
中心化后	**0.3351**	0.2354	0.3324	0.2695	0.2777
中心化前	0.2937	0.1826	0.303	0.2534	0.2809
性能提升	0.0414	**0.0528**	0.0294	0.0161	-0.0032

具体分析如下：

a）对相似度矩阵进行中心化后比中心化前的结果好，中心化相似度矩阵能够提升词向量的质量。根据表 4 和表 6，在 WS-353 数据集中，中心化前后训练得到的词向量在点积相似度和余弦相似度的结果中都是中心化后的结果好于比中心化前的结果。根据表 5 和 7，在 RW 数据集中，中心化后的结果同样好于中心化前的结果。

b）中心化后得到的词向量在点积相似度的方法下提升幅度比使用余弦相似度大。在 WS-353 数据集中，使用点积相似度计算词向量相似度时（表 4），中心化后比中心化前提高 0.2896；使用余弦相似度时（表 6），中心化后比中心化前提高了 0.2174。在 RW 数据集上，点积相似度计算词向量相似度时（表 5），中心化后比中心化前提高 0.1801；使用余弦相似度时（表 7），中心化后比中心化前仅提高了 0.0528。

c）对于词向量相似度计算方面，点积相似度更适于 WS-353 数据集，而 RW 数据集更适合余弦相似度。在 WS-353 数据集中，通过比较表 4 和 6 的中心化后训练出的词向量的结果发现，使用点积相似度的最好结果为 0.6401，使用余弦相似度的最好结果为 0.6257，点积相似度的结果好于余弦相似度的结果，且在各模型下点积相似度的结果都好于余弦相似度的结果。在 RW 数据集中，通过比较表 5 和 7 中心化后训练出的词向量的结果发现，使用点积相似度的最好结果为 0.3098，使用余弦相似度的最好结果为 0.3351，余弦相似度的结果好于点积相似度的结果，且在各模型下余弦相似度的结果都好于点积相似度的结果。

d）中心化后 GCE 模型训练出的词向量在 WS-353 数据集下表现最好，中心化后 TCE 模型训练出的词向量在 RW 数据集下表现最好。无论使用点积相似度还是余弦相似度，在 WS-353 数据集上，中心化后 GCE 都获得了最好结果，达到 0.6401 与 0.6257。而在 RW 数据集上，中心化后 TCE 获得了最好的结果，达到 0.3098 与 0.3351。

由于在 WS-353 数据集中，中心化后的 GCE 模型（GCE-

[1] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

C）的结果最好；在 RW 数据集中，中心化后的 TCE 模型（TCE-C）的结果最好。因此，使用 GCE-C 模型、TCE-C 模型、Skip-gram 模型和 Glove 模型在两个数据集下进行比较，结果如表 8 所示。从表 8 中能够看出，在 WS-353 数据集中，TCE-C 模型和 GCE-C 模型在点积相似度和余弦相似度两种方法中的结果均超过 Skip-gram 模型和 Glove 模型，其中 GCE-C 模型结果最好。在 RW 数据集中，TCE-C 模型和 GCE-C 模型在点积相似度和余弦相似度两种方法中的结果均超过 Glove 模型，TCE-C 模型的结果与 Skip-gram 模型的相当。

表 8　与 Skip-gram 和 Glove 方法比较结果

名称	WS-353 点积相似度	WS-353 余弦相似度	RW 点积相似度	RW 余弦相似度
TCE-C	**0.6325**	**0.6253**	**0.3098**	**0.3351**
GCE-C	**0.6401**	**0.6257**	0.2989	0.3324
Skip-gram	0.6229	0.6061	**0.3321**	**0.3594**
Glove	0.6098	0.6205	0.3089	0.3321

3.2.2 降维前相似度矩阵与词向量质量关系验证

为研究降维前相似度矩阵与词向量质量的关系，以 RW 数据集为例，对实验结果进一步分析。本文认为，降维前的相似度矩阵与人工标注结果越接近则质量越好，因此计算了降维前相似度矩阵与数据集中给出的人工标注结果间的 Pearson 相关系数，从而反映相似度矩阵的质量。图 3 和 4 为上述 Pearson 值与词向量质量评价结果 Spearman 相关系数（点积相似度）的对应关系；图 5 和 6 为上述 Pearson 值与降维后词向量质量评价结果 Spearman 相关系数（余弦相似度）的对应关系。根据 4 个关系图能够发现，降维前的相似度矩阵和人工标注的结果的 Pearson 相关值与词向量的质量呈线性相关，即降维前相似度矩阵与词向量的质量呈线性相关。通过对比图 3~6 中的中心化前后的关系图发现，中心化后得到的相似度矩阵与词向量的质量的线性相关性比中心化前的线性相关性强。说明在点积相似度或余弦相似度下，中心化使得相似度矩阵更符合人工标注的相似度，相似度矩阵更加合理，从而提升词向量的质量。

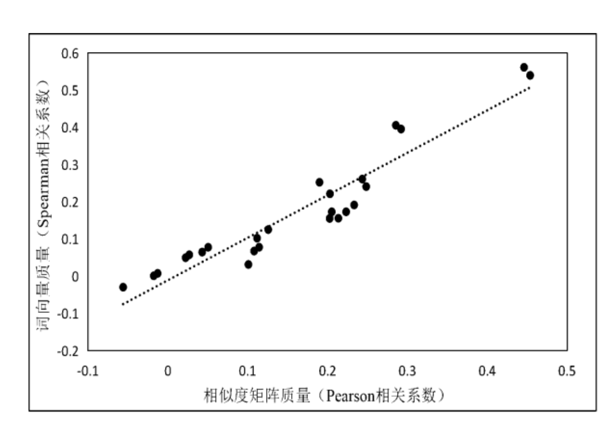


图 3　RW 数据集下，中心化前相似度矩阵质量与词向量质量间的关系（词向量相似度计算方法为点积相似度）

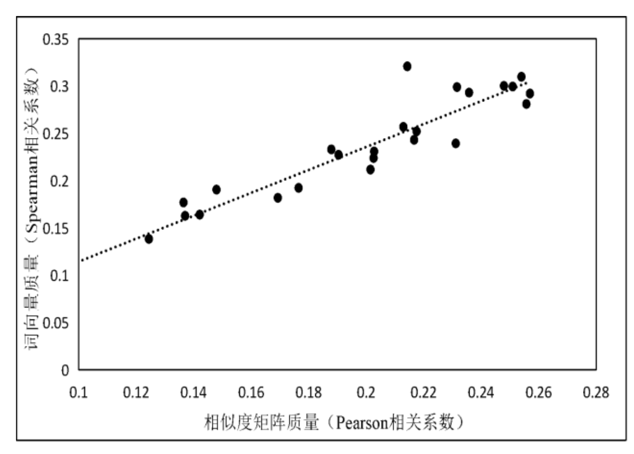


图 4　RW 数据集下，中心化后相似度矩阵质量与词向量质量间的关系（词向量相似度计算方法为点积相似度）

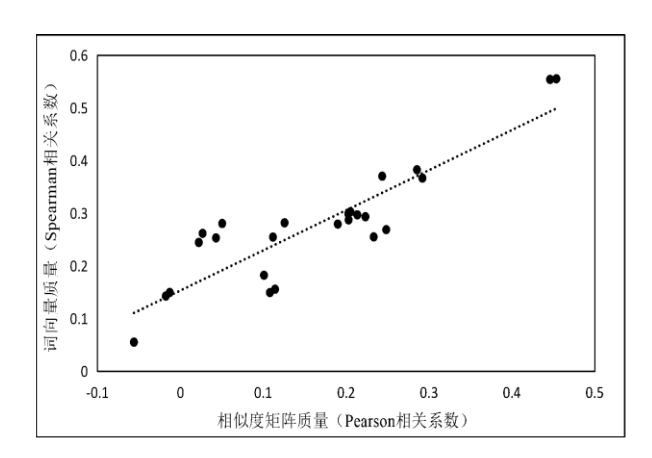


图 5　RW 数据集下，中心化前相似度矩阵质量与词向量质量间的关系（词向量相似度计算方法为余弦相似度）

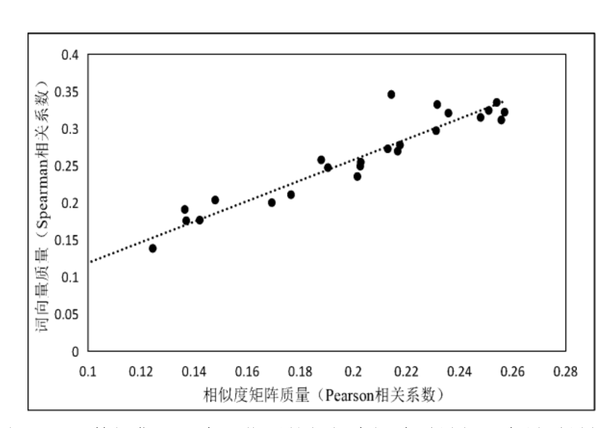


图 6　RW 数据集下，中心化后的相似度矩阵质量与词向量质量间的关系（词向量相似度计算方法为余弦相似度）

表 9 为中心化前后相似度矩阵的质量评价值（相似度矩阵和人工标注结果的 Pearson 值）与词向量质量（Spearman 值）的 Pearson 相关系数。从表 9 中能够看出，中心化后的 Pearson 值在点积相似度和余弦相似度中都高于中心化前，再次说明，中心化后的相似度矩阵更能符合人工标注的相似度结果，相似度矩阵更加合理。由此可见，中心化后能增强相似度矩阵与词向量质量的线性相关性，相似度矩阵的结果更符合人工标注的结果，相似度矩阵更加合理，从而能够提升词向量的质量。

表 9 在两种相似度计算方法下，中心化前后相似度矩阵质量与词向量质量的 Pearson 相关系数

中心化情况	点积相似度	余弦相似度
中心化后	0.9798	0.9611
中心化前	0.9518	0.8798

4 结束语

本文提出一种基于中心化相似度矩阵的词向量方法，对词-上下文共现矩阵计算出的相似度矩阵进行中心化后再进行降维得到词向量，并在 WS-353 和 RW 数据集上的词语相似性任务中验证该方法的有效性。

通过词语相似性任务的实验发现，本文提出的方法对词向量的质量有较大的提升。关键结论如下：a）降维前相似度矩阵的质量与词向量的质量线性相关，即降维前相似度矩阵与人工标注相似度越符合，词向量质量越好；b）中心化能够提高相似度矩阵质量，进而提高词向量质量。

基于以上两点结论，降维前相似度矩阵的质量是决定词向量质量的关键因素，因此构造好的相似度矩阵应是基于矩阵分解的词向量方法的工作方向。那么，若采用半监督方法对相似度矩阵的值进行部分指导，使得相似度矩阵更趋于指导信息，从而提升词向量的质量，这将是一个有趣的研究点，也是未来的主要工作。

参考文献：

[1] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning [C]// Proc of the 25th International Conference on Machine Learning. New Yokr: ACM Press, 2008: 160-167.

[2] Bengio Y, Schwenk H, Senécal J, et al. Neural probabilistic language models [J]. Journal of Machine Learning Research, 2003, 3 (6): 1137-1155.

[3] Lai S, Liu K, He S, et al. How to Generate a Good Word Embedding [J]. IEEE Intelligent Systems, 2016, 31 (6): 5-14.

[4] 于东，荀恩东．基于 Word Embedding 语义相似度的字母缩略术语消歧 [J]．中文信息学报, 2014, 28 (5): 51-59.

[5] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank [C]// Empirical Methods in Natural Language Processing. 2013: 1631-1642.

[6] Kim Y. Convolutional neural networks for sentence classification [C]// Empirical Methods in Natural Language Processing. 2014: 1746-1751.

[7] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch [J]. Journal of Machine Learning Research, 2011, 12 (1): 2493-2537.

[8] Santos C N D, Gattit M. Deep convolutional neural networks for sentiment analysis of short texts [C]// Proc of International Conference on Computational Linguistics. New York: ACM Press, 2014: 69-78.

[9] 袁书寒，向阳．词汇语义表示研究综述 [J]．中文信息学报, 2016, 30 (5): 1-8.

[10] Pennington J, Socher R, Manning C. Glove: global vectors for word representation [C]// Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[11] Deerwester S, Dumais S T, Furnas G W, et al. Richard Harshman indexing by latent semantic analysis [J]. Journal of the American Society for Information Science, 1990, 41 (6): 391-407.

[12] Caron J. Experiments with LSA scoring: optimal rank and basis [J]. History of Education Quarterly, 2001, 50 (2): 182–203.

[13] Kevin L, Curt B. Producing high-dimensional semantic spaces from lexical co-occurrence [J]. Behavior Research Methods, Instrumentation, and Computers, 1996, 28 (2): 203–208.

[14] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [C]// Proc of International Conference on Machine Learning. New York: ACM Press, 2013.

[15] Mikolov T, Yih S W, Zweig G. Linguistic regularities in continuous space word representations [C]// North American Chapter of the Association for Computational Linguistics. 2013: 746-751.

[16] Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings [J]. Bulletin De La Société Botanique De France, 2015, 75 (3): 552-555.

[17] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization [C]// Advances in Neural Information Processing Systems. 2014: 2177-2185.

[18] Li Y, Xu L, Tian F, et al. Word embedding revisited: a new representation learning and explicit matrix factorization perspective [C]// Proc of International Joint Conference on Artificial Intelligence. [S. l.] : AAAI Press, 2015: 3650-3656.

[19] Lebret R, Collobert R. Word embeddings through Hellinger PCA [C]// Proc of Conference of the European Chapter of the Association for Computational Linguistics. 2014: 482-490.

[20] Lebret R, Collobert R. Rehabilitation of count-based models for word vector representations [C]// Proc of Conference on Intelligent Text Processing and Computational Linguistics. 2015: 417-429.

[21] Turney, Peter D, Pantel, et al. From frequency to meaning: vector space models of semantics [J]. Journal of Artificial Intelligence Research, 2010, 37 (1): 141-188.

[22] Choi S S, Cha S H, Tappert C C. A survey of binary similarity and distance measures [J]. Journal of Systemics Cybernetics & Informatics, 2010, 8 (1): 43-48.

[23] Bullinaria J A, Levy J P. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD [J]. Behavior Research Methods, 2012, 44 (3): 890-907.

[24] Österlund A, Ödling D, Sahlgren M. Factorization of latent variables in distributional semantic models [C]// Empirical Methods in Natural

Language Processing. 2015: 227-231.

[25] Marina M A. Data Centering in Feature Space [C]// Proc of the 9th International Workshop on Artificial Intelligence & Statistics. 2003.

[26] 王裴岩, 蔡东风. 基于统计检验的核函数度量方法研究 [J]. 计算机科学, 2015, 42 (4): 199-205.

[27] Sedgwick P. Spearman's rank correlation coefficient [J]. British Medical Journal, 2014, 349 (nov28 1): g7327.

[28] Rivlin E. Placing search in context: the concept revisited [J]. ACM Trans on Information Systems, 2002, 20 (1): 116– 131.

[29] Luong M, Socher R, Manning C D. Better word representations with recursive neural networks for morphology [C]// Computational Natural Language Learning. 2013: 104-113.

[30] 裴楠. 基于计数模型的 Word Embedding 算法研究 [D]. 沈阳: 沈阳航空航天大学, 2017.